Docket No. 1046.1236

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yutaka HAGA

Application No.: 09/778,076   Group Art Unit: 2192

Filed: February 7, 2001   Examiner: Yigdall, Michael J.

For: APPARATUS FOR COLLECTING PROFILES OF PROGRAMS

REPLY BRIEF UNDER 37 C.F.R §§ 41.37(c)(1)(vii)

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Entry of this second Reply Brief ("second Reply Brief") is respectfully requested. This second Reply Brief is submitted in response to the Supplemental Examiner's Answer mailed March 19, 2008 ("Supplemental Examiner's Answer"), which was in response to a first Reply Brief filed on August 7, 2007 ("first Reply Brief").

An Office Communication mailed from the USPTO on October 25, 2007 ("Communication") acknowledged receipt of the first Reply Brief. An Order Returning Undocketed Appeal To Examiner ("Order") mailed from the Board of Patent Appeals and Interferences ("Board") on March 14, 2008 indicated that the Communication from the Examiner forwarding the first Reply Brief to the Board included comments by the Examiner making the acknowledgment a Supplemental Examiner's Answer. Thus, this second Reply Brief is directed, in particular, to the Supplemental Examiner's Answer, that is:

> The examiner notes that the reply brief primarily attacks the "motivation" for substituting the branch interrupts of Smolders for the timer interrupts of Alexander and appears to mischaracterize the reasoning presented in the examiner's answer mailed on June 18, 2007.

(See, Communication).

I.   STATUS OF CLAIMS

Claims 8-17, 19-28, and 30-42 are pending in this application at the filing of this second Reply Brief. Claims 8-17, 19-28, and 30-42 have at least been twice rejected. Claims 12, 23, and 34 are independent claims, and claims 8-11, 13-17, 19-22, 24-28, 30-33, and 35-42 are dependent claims.

II. GROUNDS OF REJECTION

Claims 8-17, 19-28, and 30-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Alexander, III et al. (U.S.P. 6,002,872) (Alexander) in view of Smolders (U.S.P. 6,253,338) (Smolders) and Yeh et al. (U.S.P. 6,427,206) (Yeh).

III. REPLY ARGUMENT

The Supplemental Examiner's Answer is:

> The examiner notes that the reply brief ('first Reply Brief) primarily attacks the "motivation" for substituting the branch interrupts of Smolders for the timer interrupts of Alexander and appears to mischaracterize the reasoning presented in the examiner's answer mailed on June 18, 2007.

(See, Communication).

Appellant respectfully submits that the first Reply Brief did not "mischaracterize the reasoning" presented in the Examiner's Answer. (See, for example, American Heritage® Dictionary of the English Language that defines "mischaracterize" is defined as, for example, "give a false or misleading character" at <<http://www.bartleby.com/61/10/M0331000.html>>).

Rather, Appellant submitted that the reasoning in the Examiner's Answer did not support a rejection of claims 8-17, 19-28, and 30-42 under 35 U.S.C. §103(a).

III. a) In an Amendment filed July 30, 2004, Appellant argued there was no motivation to modify Alexander to replace a timer interrupt with a branch interrupt and "a combination of the art in itself would require code modification."

Beginning on page 13 of the Examiner's Answer, the Examiner is rebutting Appellant's argument that a method including "identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code; and . . . when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated," as recited by claim 12, for example, is not taught by a combination of Alexander in view of Smolders and Yeh.

The Examiner also is rebutting Appellant's arguments that the Examiner had not provided required support for his statement that it is obvious to "substitute the timer interrupt of Alexander with the branch interrupt of Smolders."

Appellant points out that in item (9) of the Examiner's Answer, entitled Grounds of Rejection, the Examiner asserted that Alexander discloses an apparatus:

> [F]or collecting a profile of a subroutine included in a program . . . an analyzing section, when an interrupt is generated during execution of said program: obtaining a branch source address and a branch destination address from a source of said interrupt (see, for example, column 5, lines 20-32, which shows

analyzing the stack frames in response to an interrupt to identify subroutines, and column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address).

(Examiner's Answer at pages 3-4).

In the Examiner's answer, the Examiner asserted it would have been obvious:

[T]o substitute the timer interrupt of <u>Alexander</u> with the branch interrupt of molders, <u>as suggested by Alexander</u>, <u>so as to obviate any overhead and modifications to the code</u>.

(Emphasis added, Examiner's Answer at page 5).

The Examiner asserted <u>Smolders</u>' teachings:

[E]nable "tracing hardware counters by way of an interruption without introducing any overhead or modifying the code". . it would have been obvious . . . to use a branch interrupt to trigger the sampling mechanism of <u>Alexander</u>. <u>As Smolders suggests</u>, this would enable sampling without introducing any overhead and <u>without modifying the code</u> of the program.

(Examiner's Answer at page 14).

In the first Reply Brief, Appellant submitted that merely because a substitution is *arguendo* possible, in itself, is not a motivation for such a substitution, and following the Examiner's reasoning, a motivation exists for substitution of anything that could possibly be substituted.

III. b) In the first Reply Brief, Appellant also submitted that the Examiner did not correctly interpret <u>Smolders</u>' disclosure regarding modification of code, and <u>Smolders</u> does <u>not</u> teach substituting a branch interrupt for a timer interrupt does not introduce overhead nor modify the code of the program, as the Examiner asserted. Rather, Appellant pointed out that <u>Smolders</u> discloses:

Application Programming Interfaces (API) have also been built to collect counter information for portions of workloads. In this mode, <u>one must add calls to API code</u> just before and immediately after the execution of the portion of the workload to analyze. The former is to setup and start the counting and the latter is to stop the counting and retrieve the results. Those <u>API calls can either be added directly to the source code</u> if it is available or by way of instrumentation, i.e. dynamic insertion of code to divert normal execution path. . . . Consequently, it would be <u>desirable to provide</u> an improved method and system that <u>determines which part of a workload is responsible for counter increments of desired events without any overhead</u>. . . .It is yet another object of the present invention to <u>provide an improved method and system for tracing hardware counters by way of an interruption without introducing any overhead or modifying the code</u>.

(Emphasis added, col. 1, lines 33-68).

<u>Smolders</u> further discusses:

When using prior art API, lower granularity of results can be produced but only up to a certain limit determined by the overhead introduced. The <u>overhead is caused</u>

3

<u>by the counting of the API code itself. With the present invention, the code of the counter level tracing tool is not counted therefore introducing no overhead at all.</u>

(Emphasis added, col. 5, line 65 - col. 6, line 14).

Thus, Appellant submitted that <u>Smolders</u> merely teaches a way of tracing hardware counters without adding code that is <u>counted</u> as API code. The counter level tracing tool is <u>not</u> a <u>codeless</u> modification, but rather is a code/modification as described in cols 2-5 of <u>Smolders</u>, it is just that the counter level tracing tool code, itself, is not counted, i.e., no overhead, and does not modify <u>the API code, itself</u>.

Thus, Appellant submitted that <u>Smolders</u> does <u>not</u> teach that a timer interrupt can be introduced into a program without any writing of code at all, as the Examiner seems to imply and asserts as a motivation. Appellant respectfully submits that such submissions in the Reply Brief were not attempts to mischaracterize the Examiner's Answer.

III. c) In the Appeal Brief, Appellant argued that since <u>Alexander</u> clearly distinguishes a difference between different types of interrupts there is no motivation to modify <u>Alexander</u> with <u>Smolders</u>. However, in the Examiner's answer, the Examiner asserted:

> the page fault interrupt of <u>Alexander</u> is a recurring event in the <u>same sense</u> that the trace interrupt of <u>Smolders</u> is a recurring event. While another interpretation of a "periodic event" is an event that occurs at a fixed frequency or interval, one of ordinary skill in the art would recognize that the page fault interrupts of <u>Alexander</u> are not necessarily generated at a fixed frequency or interval. Thus, a conclusion that the sampling mechanism of <u>Alexander</u> is strictly limited to interrupts that are fixed-frequency events is not supported. The discussion of the page fault interrupt in <u>Alexander</u> is, at minimum, evidence in-favor of a reasonable expectation of success in substituting the timer interrupt with another recurring event such as <u>Smolders</u>' trace interrupt.

(Examiner's Answer at page 15).

In the Reply Brief, Appellant argued that the Examiner was attempting to establish motivation by merely arguing a substitution is *arguendo* possible. However, Appellant in reply pointed out that <u>Alexander</u> clearly <u>distinguishes</u> between the utility and function of the timer fault and the page fault. In particular, <u>Alexander</u> discloses:

> The depicted example illustrated in FIGS. 2 through 10 illustrated a process for structured profiling of data processing systems and applications executing on a data processing system. The depicted example in these figures <u>is directed</u> towards obtaining information from a stack in <u>response to an occurrence of a timer interrupt</u>.

(See, for example, col. 8, lines 40-50).

<u>Alexander</u> further teaches <u>functional differences</u> between interrupts and different embodiments of <u>Alexander</u>'s invention as:

> The processes of the present invention also may be applied to <u>other types of</u>

4

events occurring within a data processing system or application. For example, in another embodiment of the present invention, the processes described may be applied to memory analysis. Instead of employing a timer interrupt, a page fault interrupt may be used as a signal to gather data from the stack. In the depicted example, a page fault is forced each time a memory access occurs within the data processing system. A page fault also may be selectively triggered, depending on processing resources.

(col. 8, starting at line 48)

Since Alexander teaches different interrupts may be used for different purposes, Alexander does not teach a blanket substitution of the timer interrupt with a page fault, as the Examiner appears to imply. Thus, Appellant submitted it is not obvious in view of Alexander that interrupts can be substituted for "collecting a profile of a subroutine."

Further, Appellant submitted that the assertion by the Examiner that interrupts can be substituted is an unsupported conclusory statement and is further contradicted by the art relied on by the Examiner.

The Examiner further asserted it would have been obvious

to supplement the profiling system of Alexander with the features taught by Yeh and to identify the type of branch, as taught by Yeh, so as to collect branch predictions and enhance the collection of profiles for purposes of speculative execution.

(Examiner's Answer at page 5).

In the Reply Brief, Appellant pointed out that claim 12 recites "obtaining a branch source address and a branch destination address from a source of said interrupt," and if Alexander was modified to include a page fault, as the Examiner suggested then other features of claim 12 would not be taught by the cited art, since a source of the interrupt would not be a branch source address and a branch destination address, but something else.

Further, Appellant pointed out that claim 12 recites "when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (emphasis added)." However, the Examiner asserted:

when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (see, for example, Yeh, step 310 in FIG. 3 and column 8, lines 42-48, which shows not collecting history information when the branch is not a calling instruction or a return instruction, and see, for example, Smolders, steps 46 and 64 in FIG. 3 and column 5, lines 43-52, which shows terminating the interrupt when trace information is not to be collected).

(Examiner's Answer at page 6).

Again, the Examiner appeared to establish motivation by merely asserting a modification is arguendo possible. Thus, Appellant respectfully pointed out that in KSR International Co. v. Teleflex Inc., 82 USPQ2d 1385, 127 SCt 1727, 167 LEd2d 705 (U.S. 2007), the U.S. Supreme

Court held that in determining obviousness, one "must ask whether the improvement is more than the predictable use of prior art elements according to their established functions" slip op. 13, 82 USPQ2d at 1396 and it is necessary "to determine whether there was an apparent reason to combine the known elements in the fashion claimed" slip op. 14, 82 USPQ2d at 1396.

The Supreme Court further affirmed *In re Kahn,* 441 F. 3d 977, 988 (CA Fed. 2006), stating: "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."

III. d) Starting on page 17 of the Examiner Answer, the Examiner is rebutting Appellant's argument that claim 40 recites an apparatus including names and labels given to the tables in a certain relationship that render the same "functionally distinct" from an arbitrary association of pointers and nodes referred to by the Examiner by asserting that the claim 40 recited subject matter "amounts to a data structure that includes arbitrarily named tables."

Appellant submitted that the data structures are not "arbitrarily" named, Rather, claim 40 recites an apparatus including a "table that is generated corresponding to each executor of the subroutine . . . executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table . . . a pointer to specify the subroutine managing table managing the calling subroutine. (emphasis added)." That is, claim 40 recites features that are functionally interrelated with the names of features indicating, in part, such a functional relationship and actions. However, the Examiner asserted:

> One of ordinary skill in the art would recognize that such pointers are necessary to allow the elements of the data structure to be located, accessed, traversed and so on. . . . appreciate that tree structure 500 may be implemented in a variety of ways and that many different types of statistics may be maintained at the nodes other than those in the depicted example. In addition, other pointers may be stored within the nodes to further aid subsequent analysis. Further, other structural elements, such as tables for properties of the routine. . . . it would have been obvious ... to implement the data structure of Alexander in a form suitable for subsequent analysis. . . .Alexander suggests that one of ordinary skill in the art might choose to implement the data structure in any form suitable for the subsequent analysis that he or she intends to perform.

(Examiner's Answer at page 18).

Appellant respectfully submitted that using the Examiner's logic that since pointers are "necessary," and trees can be implemented in a variety of ways - - any recited relationship between one table and another, including generation of a specific is obvious in view of Alexander, regardless of the implementation, thus Appellant argued that the Examiner assertions are attempting to establish motivation by merely arguing a modification is *arguendo* possible.

6

Since the Examiner did not provide support why such a modification of <u>Alexander</u> "to implement the data structure in any form suitable" should be accomplished, Appellant respectfully submitted that the Examiner's answers.are mere conclusory statements, and as stated by the Supreme Court *In re Kahn* "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements."

IV.   CONCLUSION

Appellant respectfully submits that the first Reply Brief did not "mischaracterize the reasoning" presented in the Examiner's Answer.

In view of the law and facts stated herein, the Appellant respectfully submits that the Examiner has failed set forth a *prima facie* obviousness case against the pending claims.

For all the foregoing reasons, the Appellant respectfully submits that the cited prior art does not teach or suggest the presently claimed invention. The claims are patentable over the prior art of record and the Examiner's findings of unpatentability regarding claims 8-17, 19-28 and 30-42 should be reversed and the patentability over the presently cited references be affirmed.

The Commissioner is hereby authorized to charge any additional fees required in connection with the filing of this second Reply Brief to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: _May 14, 2003_

By _Paul W. Bobowiec_
Paul W. Bobowiec
Registration No. 47,431

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501